Descriptif de cours de spécialité

Sup Galilée INFORMATIQUE

Année 1

U.E. Outils Mathématiques et Algorithmiques

Voir description des cours communs scientifiques

Algorithmique (cours : 19.5h, TD : 39h) ALG
• Notion d'algorithmes, calculs de complexité (exemples avec les algorithmes de tri) • Programmation dynamique • Algorithmes gloutons • Quelques algorithmes particuliers : Karatsuba, Strassen, Rabin-Karp, Boyer-Moore • Notions de NP-complétude, réductions simples.

U.E. Programmation

Informatique de base (cours : 10.5h, TP : 21h) INB
• Eléments de syntaxe • Structure d'un programme • Structure de contrôle • Fonction, variable, durée de vie • Gestion de la mémoire • Types de données (prédéfinis, pointeurs, énumération, structure) • Chaînes de caractères • Bibliothèques • Programmation modulaire • Gestion des entrées/sorties

Unix (cours : 12h, TD : 12h) UNX
• Généralités sur le système Unix • Introduction au système de fichiers (disques, répertoires, arborescence, dénomination) • Quelques commandes de manipulation de fichiers et de répertoires (création, copie, déplacement, changement de nom, suppression) • Les processus • Les interpréteurs de commandes

Génie logiciel – 1 (cours : 12h, TD : 12h) GL1
• Travail à partir d'études de cas dont il faut identifier les structures de données • Ecriture modulaire de spécifications simples ou instances de génériques

Architecture des Ordinateurs (cours : 19.5h, TD : 27h, TP : 12h) ARC
• Le jeu d'instruction de MIPS : instructions, formats, modes d'adressage • Le co-processeur flottant • La hiérarchie de la mémoire : caches, mémoire virtuelle • Circuits logiques, contrôle et assemblage du chemin de données • Parallélisme d'instructions • Analyse de performance • Quelques notions sur les entrées-sorties et les interruptions

Système d'exploitation (cours : 19.5h, TD : 19.5h, TP : 19.5h) SYS
• Système de fichiers • Gestion des processus • Les threads • Communication inter-processus • Synchronisation des processus • Interblocage • Ordonnancement des processus • Gestion de la mémoire • Mémoire virtuelle

U.E. Outils formels

Sémantiques des langages de programmation (cours : 13.5h, TD : 13.5h) SEM
• Introduction. Définition d'un langage "while" d'exemple • Sémantique opérationnelle: sémantique naturelle, sémantique opérationnelle structurelle. • Implémentations prouvablement correctes: machines abstraites, correction. • Sémantique dénotationnelle: style direct, théorie des pointes fixes.

Modélisation (cours : 19.5h, TD : 19.5h) MOD
• Modélisation en langage UML • Propriétés exprimées en OCL • Structuration à partir des schémas de problème (Problem Frames) et de schémas de diagrammes UML associés aux schémas de problème

Logique (cours : 10.5h, TD : 21h) LOG
• Calcul des propositions : syntaxe, sémantique, preuves (calcul des séquents), théorème d'adéquation. • Calcul des prédicats : syntaxe, sémantique, preuves (calcul des séquents), théorème d'adéquation.

Théorie des langages (cours : 15h, TD : 15h) TL
• Automates à états finis et langages rationnels ou réguliers • Expressions rationnelles ou régulières. • Correspondance entre expressions rationnelles et automates à états finis. Lemme de pompage ou de l'étoile. • Langages non rationnels. Minimisation des automates. • Grammaires syntagmatiques. Hierarchie de Chomsky. Grammaire contextuelles et décidabilité. Grammaires linéaires et langages rationnels. • Grammaires et langages hors-contexte ou CF. Analyseurs déterministes. Normalisation des grammaires CF. Automate a pile et langages déterministes. Parsage tabulaire.

U.E. Outils et projets

Projet (TP : 40h) PROJ
• Introduction au cycle en V du développement de logiciel. • Phase de spécification. • Phase de conception. • Phase d'implémentation et mise au point (campagne de tests). • Livraison du logiciel et soutenance.

Infographie (cours : 12h, TD : 12h, TP : 24h) BD
• Infographie 2D : Tracé de segments, cercles et polygones. Remplissage de polygones, fenêtrage de polygones. • Infographie 3D : Notion sur les maillages, maillage du cube, du cylindre, de la sphère, tranalation et rotation de maillage. Affichage elementaire d'un maillage, detection des partie cachée (algorithme du z-buffer), lumière et texture.

Base de données (cours : 18h, TD : 18h, TP : 18h) BD
• Modélisation Entité Association • Modèle relationnel, dépendances fonctionnelles et normalisation • Algèbre relationnelle • SQL : Langage de définition de données (création et mise à jour du schéma d'une base de données contraintes, dictionnaire des données, peuplement d'une base de base de données) et langage de manipulation de données (requêtes SQL simples et imbriquées, définition et manipulation de vues).

Programmation Orientée Objet (cours : 10.5h, TD : 10.5h, TP : 15h) POO
• Type primitifs de Java • Structure d'un programme java • Les opérateurs et les expressions • Les instructions de contrôle • Les classes et les objets • Les tableaux • L'héritage • Gestion des exceptions • Les entrées/sorties • Les threads • Les collections et algorithmes • Programmation générique

Stage découverte entreprise STA1
• Stage d'un mois non technique de découverte de l'entreprise. Ce stage est évalué par une soutenance et un rapport.

Année 2

U.E. Algorithmique

Algorithmique de Graphes (cours : 18h, TD : 18h) GRAP
Modéliser des problèmes par des graphes et les résoudre algorithmiquement • Introduction • Notions élémentaires et définitions • Représentation des graphes • Parcours des graphes • Problèmes de meilleurs chemins • Arbres couvrants • Flots dans les graphes

Programmation Linéaire (cours : 18h, TD : 18h, TP : 12h) PL
La théorie de la programmation linéaire et les algorithmes de résolution • Introduction • Modélisation et définitions • Rappels d'algèbre linéaire • Algorithme primal du simplexe • Algorithme dual du simplexe • Dualité • Analyse de sensibilité

Programmation Fonctionnelle (cours : 15h, TD : 15h, TP : 15h) PFONC
Un paradigme de programmation non impérative, la programmation fonctionnelle, et ses bases théoriques, le lambda-calcul • Bases de lambda-calcul • Algorithme d'inférence de types (Hindley-Milner) • Langage CAML • Transformations et preuves de programmes • Algorithmes de ramasse-miettes

U.E. Outils

Génie Logiciel Avancé (cours : 18h, TD : 18h) GLA
• Descriptions structurées de cas d'utilisation UML • Méthode de spécification formelle des systèmes dynamiques s'appuyant sur la description des cas d'utilisation • Méthode de spécification des propriétés des systèmes dynamiques • Réseaux de Petri places-transitions & méthode de spécification • Introduction aux réseaux colorés

Programmation objet C++ (cours : 18h, TP : 18h) C++
La programmation objet et le langage C++ : • Du C au C++ (évolution du C, différences pour les fonctions, approche objet en C++) • Classes et instances (données et méthodes d'instances et de classes, autoréférence, constructeurs et destructeurs, constructeur de copie, fonctions amies) • Surcharge des opérateurs (principes généraux et cas particuliers) • Dérivation et héritage (contrôle des accès, constructeurs et destructeurs dans les dérivations, dérivations multiples, conversions et dérivations, méthodes virtuelles et classes abstraites, polymorphisme) • Flux et entrées/sorties (classes définissant les flux, flux et fichiers, personnalisation des flux) • Modèles de fonctions et modèles de classes (généricité en C++ des classes et des fonctions, surcharges des modèles de fonctions) • Gestion des anomalies (notion d'exception et approche objet pour la gestion des erreurs, discrimination et remontée d'exceptions)

Programmation Logique (cours : 15h, TD : 15h, TP : 15h) PLOG
• Rappels logique propositionnelle, résolution propositionnelle, correction et complétude par réfutation • Rappels logique d'ordre un, forme normale conjonctive, unification, résolution, correction et complétude par réfutation • Le langage PROLOG (modèle logique, contrôle non logique, négation, …) • Introduction à l'Intelligence Artificielle (recherche dans les arbres de jeux), modélisation et implémentation en PROLOG d'un jeu et d'un algorithme de recherche dans un arbre de jeu.

Web sémantique et IA (cours : 18h, TP : 18h) WSEM
Les recommandations du W3C, les enjeux du WS • XML / DTD , XSD, RDF/RDFS, SPARQL, OWL, • Utilisation pratique en ligne et hors ligne

U.E. Fondamentaux

Optimisation Combinatoire (cours : 18h, TD : 18h, TP : 12h) OC
Modéliser des problèmes en termes de Programmes Linéaires en Nombres Entiers (PLNE), savoir calculer des bornes inférieures et supérieures à la solution optimale et pour des cas simples savoir trouver la valeur optimale et la solution optimale. • Définitions et Modèles; • Dualité et application au problème du transport • Relaxations : linéaire, lagrangienne, agrégée, etc. • Méthode de séparation et évalutation • Méthode dite « cutting planes » • Programmation Dynamique • Heuristiques : gloutonne, lagrangienne et meta-heuristiques

Réseaux 1 (cours : 18h, TD : 13.5, TP : 8) RES1
• Introduction aux réseaux -Définition de base, types de réseaux -Hiérarchie protocolaires, le modèle en couches OSI, le modèle TCP/IP • La couche physique -Médiums de transmission, topologies -Codages, modulations, numérisation • Le sous-couche MAC -Multiplexages temporel et fréquentiel, - Les méthodes d'accès aléatoires : CSMA/CD et CSMA/CA • La sous-couche gestion de liaison de données -Détection et correction des erreurs, mécanismes de contrôle de flux -Exemple d'un protocole liaison : HDLC • Réseaux Ethernet et WiFi • Interconnexion des réseaux locaux -Le spanning tree -ARP, -Introduction à IP : adressage, sous-réseaux, routage direct.

Base de Données Avancées (cours : 18h, TD : 12h, TP : 12h) BDA
• Développement d'applications avancées de Bases de Données : • SQL avancé • PL/SQL • UML et SQL2/SL3 • Intégrité, Sécurité des Données, et Administration • Optimisation • DataWareHouse

Ingénierie des Connaissances (cours : 18h, TP : 18h) IC
• Historique de la modélisation des connaissances en IA. Les systèmes experts • Acquérir et modéliser des connaissances : les différentes approches • L'acquisition de connaissances à partir de textes: Problématiques, enjeux, méthodes et outils • Formaliser des connaissances : Logiques du premier ordre, réseaux sémantiques, logiques de description • Les ontologies et le web sémantique : Historique de la notion d'ontologie. Développement d'exemples montrant les problèmes autour de la construction d'ontologies. Présentation d'applications utilisant ces ontologies dans le cadre du web sémantique. • Puis 4 séances de TD sur machines visant à faire manipuler un logiciel de construction d'ontologies à partir de textes et à contruire une ontologie de domaine.

Administration Systèmes (cours : 18h, TP : 18h) ADSYS
• Installation et configuration de stations de travail et de serveurs • Gestion de systèmes de fichiers • Procédures d'amorçage de système, • Installation de logiciels • Administration de la sécurité et des utilisateurs • Sauvegardes et reprises du système • L'installation et la configuration de différents services sont passées en revue: gestion centralisée des utilisateurs (NIS, NIS+, LDAP), résolution d'adresses (DNS), messagerie, système de fichiers en réseau (NFS, Samba), service d'impression, serveurs de temps, sauvegarde (bandes, réseau de sauvegarde), mise en place de pare-feu, etc.

Interfaces graphiques (cours : 12h, TP : 24h) IG
Développement d'interfaces graphiques avec Java Swing : • Rappels Java (classes, contrôle des accès, classes internes, méthodes anonymes, flots d'entrée-sortie) • Construction d'interfaces graphiques et fenêtrées avec Swing (objets primitifs et gestionnaires d'affichage, menus, barre de menus, barres d'outils, dialogues) • Développement d'applications graphiques (comment dessiner) • Eléments de contrôle, disposition et présentation • Gestion des événements • Gestion d'animations (dessins, images et threads d'animation).

U.E. Projets

Projet PR
• Aborder une réalisation dans un domaine éventuellement nouveau, développer le projet depuis la description UML, jusqu'au codage et test. Présenter le projet dans un rapport et dans une présentation orale. Travail en équipe. • Comprendre le projet demandé (reformulation), l'analyser, évaluer les difficultés techniques, donner une description UML, puis fournir une architecture de conception, de détail et le code. • Fournir un manuel utilisation, un manuel d'installation. • Rédiger un rapport complet

Stage Technicien STA2
• Stage de deux à trois mois. Ce stage doit contenir un travail de développement informatique, en équipe ou en groupe. L'étudiant travaille le plus souvent à partir d'une spécification préalable, qu'il peut être amené à faire évoluer. Idéalement, ce stage comprendra également une étape de test/validation et la rédaction d'une documentation utilisateur. Ce stage est évalué par une soutenance et un rapport.

Année 3

U.E. Tronc Commun

Sécurité et Cryptographie (cours : 18h, TD : 18h) CRYPT
• Politiques et modèles de sécurité informatique; • Généralités sur les procédés de chiffrement et outils mathématiques pour la cryptographie; • Procédés de chiffrements à clef secrète classiques; • Notions d'analyse fréquentielle des textes chiffrés; • Théorie de l'information de Shannon (sécurité inconditionnelle et entropie d'un cryptosystème); • Généralités sur les procédés de chiffrement par blocs itérés (réseau de Feistel); • Descriptions détaillées des standards modernes de cryptographie actuellement employés DES, IDEA, AES; • Généralités sur les cryptosystèmes à clef publique; • Description détaillée du procédé de chiffrement à clef publique RSA.

Gestion des réseaux informatiques (cours : 15h, TD : 12h, TP : 18h) RES2
Cours : • Introduction • La couche Réseau • Le routage dans l'Internet (RIP, OSPF) • La couche Transport • La couche Application • Les services Réseaux (DHCP, NAT, Firewall) TPs : Configuration de routeurs CISCO 1841 pour : • Mettre en place un routage Statique • Mettre en place un routage RIP • Mettre en place un routage OSPF • Mettre en place un serveur DHCP, NAT et Firewall

Compilation (cours : 15h, TD : 15h) COM
• Introduction et discussion des phases de compilation d'un langage de programmation. • Analyse lexicale. Langages rationnels. Outils pour la génération automatique d'un analyseur lexical (lex ou flex). • Analyse syntaxique. Grammaires hors-contexte. Analyse descendante et analyse ascendante. Précédence des opérateurs. Parsage LR. Outils pour la génération automatique d'un analyseur syntaxique (yacc ou bison). • Attributs et traduction directe par la syntaxe. Analyse sémantique. • Infrastructure d'exécution. Table des symboles. Organisation de la mémoire. • Génération de code. Code intermédiaire et code cible. Optimisation de code.

Programmation Web Avancée (cours : 10.5h, TD -TP : 18h) PWA
• Structure et composants d'un service Web • Librairies pour l'Analyse du XML (DOM, SAX, JAXP) • Librairies pour client/serveur Web (Javascript, PHP, AJAX, JSP) • Environnements de déploiement (J2EE, JSP, Beans, Hibernate, etc.)

Traitement Informatique des Données (cours : 18h, TD-TP : 18h) TID
• Les méthodes descriptives : analyse en composantes principales, analyse des correspondances, méthodes de classification ou de typologie. • Les méthodes explicatives : régression multiple, analyse canonique, analyse discriminante. • Les méthodes prospectives : apprentissage à partir de données, minimisation du risque empirique (ERM), Consistance de l'approche ERM, dimension VC, minimisation du risque structurel (SRM), algorithmes génétiques, réseaux de neurones artificiels.

Algorithmique Parallèle et Distribuée (cours : 9h, TD-TP : 9h) APD
• Le modèle PRAM (Parallel Random Access Machine) ; • La parallélisation automatique ; • La programmation : MPI et OpenMP; • Éléments de parallélisme dans le processeur : lecture anticipée • L'ordonnancement ; • L'algorithmique distribuée

Projet de fin d'études PFE
• Le but de ce projet est pour les étudiants de développer un projet industriel en collaboration et de produire et présenter l'esquisse d'une solution.

Stage Fin d'études STA3
• Confier la résolution d'un problème concret à un élève ingénieur formé aux disciplines classiques d'une grande école d'ingénieur. Eprouver, au travers du projet, les qualités professionnelles et personnelles de l'élève. Rédaction d'un rapport complet et présentation du travail effectué lors d'une soutenance.

U.E. Option Communication Homme Machine et Documents Electroniques

Hypertextes Adaptatifs (cours : 6h, TP : 9h) HYP
• Notions générales d'adaptation (adaptabilité, adaptativité) dans les hypermédia • Cas de la personnalisation sur Internet • TP sur des flux RSS adaptatitifs en web 2.0

Traitement des Données Vocales (cours : 7h, TD : 7h, TP : 7h) VOC
• Production du signal sonore • Perception et psychoacoustique • Codage, compression • Formats et supports • Pré-traitement du signal de parole • Traitement du son • Spatialisation • Indexation • Emotion • Reconnaissance • Synthèse • Biométrie

Traitement des Données Visuelles (cours : 7h, TD : 7h, TP : 7h) VIS
• Rappels de traitement de signal et d'estimation : acquisition, échantillonnage,filtrage, signaux aléatoires, estimation spectrale. • Application aux données, en particulier images et signaux biomédicaux : détection et restauration de signaux biologiques, traitement d'images médicales, reconstruction tomographique, reconnaissance de formes, compression de signaux biologiques

Traitement Automatique des Langues (cours : 15h, TD : 5h, TP : 12h) TAL
• Présentation des techniques symboliques et numériques d'analyse du contenu textuel : – bases de statistiques textuelles ; – formalismes et méthodes pour l'analyse morphologique, syntaxique et sémantique ; – notions de pragmatique et de dialogue pour la communication homme-machine. Les notions sont abordées en fonction des paliers d'analyse linguistiques mais l'étude de quelques applications phares montre leur complémentarité.

Techniques de Manipulation de Documents Textuels (cours : 12h, TD : 12h) TMDT
• Introduction à Perl • Analyse de documents textuels spécialisés • Extraction d'entités nommées • Segmentation en mots et en phrases • Analyse morpho-syntaxique (Etiquetage morphosyntaxique, lemmatisation) • Extraction de termes • Acquisition de relations entre termes A cette occasion, les étudiants sont amenés à manipuler des outils de TAL existants (la plupart du temps des prototypes issus de laboratoire de recherche) et/ou à implémenter des algorithmes.

Gestion Electronique de Documents (cours : 9h, TP : 6h) GED
• Formats de documents électroniques • Systèmes d'édition de documents électroniques • Notions approfondies de XML (balisage, DTD, feuille de style, parser, DocBook et TopicMaps, XSL-FO).

U.E. Option Aide à la Décision et Optimisation

Aide à la décision (cours : 18h, TP : 6h) AD
• Formulation par la Programmation Linéaire de problèmes de planification optimale (production, couverture de charge, décisions d'achats) • Formulation par la Programmation Linéaire en Nombres Entiers de problèmes de localisation optimale (études de cas) • Structuration de problèmes et résolution par Solveur Excel (TP) • Ordonnancement de projet : construction du graphe, calcul des calendriers de dates au plus tôt, au plus tard, marges totales et libres, tâches critiques, pilotage et diagrammes de Gantt • Définition des problèmes multi-critère et multi-objective, définition de région de Pareto et compréhension de la fonction d'utilité ou fonction valeur • Problème multi-objectifs : les différentes méthodes pour trouver la région de Pareto dans le cas générale et en particulier pour le problème connu sur le nom de MOCO (Multi Objective Combinatorial Optimization) • Problème multi-critères : leurs application et les méthodes pour les traiter : AHP (Analityc Hierarchy Process), Electre I, II et III et Naiade • Résolution des exercices a l'ordinateur pour des problèmes continus comme le problème du sac à dos et le set covering et sur des problèmes discrets

Optimisation et Logiciels (cours : 9h, TP : 12h) OPL
• Présentation et utilisation de solveurs (Cplex, X-Press) et de modeleurs (OPL, AMPL) du commerce. • Présentation et utilisation de solveurs (COIN-OR, Glpk, LP-solve, Abacus) et modeleurs (OptimJ) du domaine libre. • Comparaison de l'utilisabilité et de la ré-utilisabilité, de la performance et de l'interactivité (saisie, visualisation) de ces outils logiciels

Planification et Contraintes (cours : 12h, TP : 9h) PC
• Modèles et méthodes de résolution de problèmes : ordonnancement, tournées, localisation, gestion de la chaîne logistique (Supply Chain) • Introduction à la Programmation Par Contraintes (PPC) : - Choco: un système de PPC écrit entièrement en Java, disponible en open-source et supporté par KLS OPTIM. - CHIP: un système de PPC développé par COSYTEC. • Travaux pratiques sur des exemples d'applications issues de l'industrie • Stratégies et heuristiques pour le calcul de solutions de qualité.

Résolution de problèmes de grande taille (cours : 15h, TP : 6h) RPGT
• Problèmes de grande taille : Décomposition de Dantzig-Wolfe, génération de colonnes, application au problème de découpe industrielle (*Cutting Stock Problem*) • Métaheuristiques : heuristiques gloutonnes, heuristiques de réparation, amélioration locale, métaheuristiques monosolution (méthode de recherche tabou, méthode de recherche à voisinages variables, recuit simulé), métaheuristiques basées sur une population de solutions (algorithmes génétiques et recomposition de chemins)

Etude de cas (cours : 30h) EC
• Conférences Industrielles: – Air Liquide – GDF Suez – IBM – SNCF – Eurodécision – Orange Labs – Schneider-Electric Industries – EDF